Computing and Informatics, Vol. 30, 2011, 447–466

CRYPTANALYSIS OF CIKS-128 AND CIKS-128H SUITABLE FOR INTELLIGENT MULTIMEDIA AND UBIQUITOUS COMPUTING SYSTEMS

Changhoon Lee

School of Computer Engineering
Hanshin University, Republic of Korea
e-mail: chlee@hs.ac.kr

Jongsung Kim*

Division of e-Business
Kyungnam University, Republic of Korea
e-mail: jongsungk@kyungnam.ac.kr

Jaechul Sung

Department of Mathematics
University of Seoul, Republic of Korea
e-mail: jcsung@uos.ac.kr

Yang-Sun Lee

Department of Information and Communication Engineering
Chosun University, Republic of Korea
e-mail: yslee48@gmail.com

Chang Hoon Lee

Department of Computer Engineering
Hankyong National University, Republic of Korea
e-mail: be4u@hknu.ac.kr

* corresponding author

Abstract. Recently, data-dependent permutations (DDP) that are very suitable for intelligent multimedia and ubiquitous computing systems have been introduced as a new cryptographic primitive for the design of fast encryption systems. The CIKS-128 and CIKS-128H block ciphers are the typical examples of DDP-based encryption algorithms. In this paper, we show that CIKS-128 and CIKS-128H are vulnerable to related-key differential attacks. We first describe how to construct their full-round related-key differential characteristics with high probabilities and then we exploit them to break the full-round CIKS-128 and CIKS-128H with 2^{44}, and 2^{48} data/time complexities, respectively.

Keywords: Intelligent multimedia and ubiquitous computing systems, encryption algorithms, CIKS-128, CIKS-128H, related-key differential characteristics

1 INTRODUCTION

One of the most important issues in the field of intelligent multimedia and ubiquitous computing systems is on how to protect sensitive data and how to offer data and entity authentications. The most common method to solve this security problem is to use encryption algorithms such as block ciphers, stream ciphers and public-key ciphers. However, the intelligent multimedia and ubiquitous computing systems require low power devices with high speeds, which implies that light and efficient encryption algorithms can only be used in such systems.

Recently, data-dependent permutations (DDP), which can be easily embedded in microcontrollers and general purpose CPUs, have been introduced as one of cryptographic primitives suitable to attain such a goal. Several DDP-based ciphers have been proposed for hardware implementation with low cost, such as CIKS-1 [17], SPECTR-H64 [3], and Cobra-S128 [5]. Since all of them use very simple key schedules in order to have no time consuming key preprocessing, they are suitable for the applications of many networks requiring high speed encryption in the case of frequent change of keys. Actually, they are competitive with the other well used encryption algorithms, such as AES, for different variants of hardware implementation [19, 21].

However, they have two fatal weaknesses in security. One is that the CP (controlled permutation) boxes, which are a sort of the DDP, are cryptographic linear primitives in which there exists the linearity of the sum of their output bits. By using this property [13] resulted in a linear attack on CIKS-1 faster than exhaustive search. Nevertheless, most of the DDP-based ciphers are secure against linear cryptanalysis due to the use of additional components with small non-linearity. The other is simply designed key scheduling algorithms, which make the cryptanalysts apply the related-key attack easily to such kinds of block ciphers [10, 11, 14]. Indeed, the related-key attack is one of useful tools for evaluating the security of block ciphers. Though it is hard to mount in general, this kind of attack can be practical in some of

the current real-world applications such as the IBM 4758 cryptoprocessor [22], PGV-type hash functions, message authentication codes, recent authenticated encryption modes [23], cases of key-exchange protocols that do not guarantee key integrity, and key-update protocols that update session keys using a known function [8].

CIKS-128 [3] and CIKS-128H [20], which are the improved versions of CIKS-1 and SPECTR-H64 [4], are 128-bit block ciphers with a 256-bit key size. Specially, CIKS-128H uses non-linear controlled elements (NCE) instead of CP boxes in order to be free of any linearity. These ciphers also have better hardware implementations (FPGA and ASIC) than other ciphers used in security layers of most of wireless protocols, WAP, OMA, UMTS, IEEE 802.11 and so on.

In this paper, we first present the structural properties of the controlled permutations used in the round functions of two ciphers, which allow us to make full-round related-key differential characteristics with high probabilities. We then present new related-key differential attacks on the full-round CIKS-128 and CIKS-128H which require about 2^{44} and 2^{48} data/time complexities, respectively. Our attacks correct and improve the results in [11].[1] Table 1 summarizes our results.

Block Cipher	Number of Rounds	Complexity Data/Time
CIKS-128	12 (full)	2^{44}RK-CP/2^{44}
CIKS-128H	8 (full)	2^{48}RK-CP/2^{48}

RK-CP: Related-Key Chosen Plaintexts, Time: Encryption units

Table 1. Summary of our related-key differential attacks on CIKS-128 and CIKS-128H

This paper is organized as follows. In Section 2, we briefly present how the related-key differential attack works in block ciphers. Section 3 describes two block ciphers CIKS-128, CIKS-128H, and Section 4 gives their structural properties. Sections 5 and 6 present related-key differential characteristics and key recovery attacks of CIKS-128 and CIKS-128H, respectively. Finally, we conclude in Section 7.

2 THE RELATED-KEY DIFFERENTIAL ATTACK

The related-key differential attack was firstly introduced in 1992 by Knudsen [12], and it was later formalized and generalized in 1993 by Biham [1]. The attack method applies differential cryptanalysis [2] to the cipher with different, but related unknown keys. The key relation between the related keys is known to the attacker, but the related keys themselves are not revealed. This attack is based on the key scheduling algorithm and on the encryption/decryption algorithms, hence a cipher with a weak key scheduling algorithm may be vulnerable to this kind of attack.

[1] [11] presented two attacks on CIKS-128 and CIKS-128H which are composed of wrong $P_{8/12}^{-1}$ and $R_{8/12}^{-1}$, respectively, and thus they do not work. In this paper, we present new attacks on the correct CIKS-128 and CIKS-128H.

We assume that an n-bit block cipher $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ is divided into $E = E^f \circ E^0$, denoted $E_K = E^f_K(E^0_K(P))$, where P is an n-bit plaintext, K is a k-bit secret key, and E^0, E^f and E are all permutations on n bits for each k-bit secret key. The related-key differential attack exploits a related-key differential characteristic $\alpha \to \beta$ for E^0 with probability p larger than 2^{-n}, i.e.,

$$Pr_{X,K}[E^0_K(X) \oplus E^0_{K \oplus \Delta K}(X \oplus \alpha) = \beta] = p > 2^{-n},$$

where ΔK is a non-zero key difference chosen by the attacker. Then we can retrieve the subkey of E^f as follows:

1. Collect about $c \cdot p^{-1}$ plaintext pairs (P, P') whose differences are all α, where $c > 1$.
2. Obtain the corresponding ciphertext pairs (C, C') using the keys $(K, K \oplus \Delta K)$.
3. For each subkey candidate k for E^f, calculate the number of (C, C') pairs satisfying

$$D^f_k(C) \oplus D^f_{k \oplus \Delta k}(C') = \beta, \tag{1}$$

where $D^f_k = (E^f_k)^{-1}$, $D^f_{k \oplus \Delta k} = (E^f_{k \oplus \Delta k})^{-1}$ and Δk is the subkey difference of E^f derived from the key difference ΔK (if the key schedule is linear, then Δk is uniquely determined by ΔK). We denote this number by T_k. Output subkey k as the right key of E^f if all $T_k > T_{k'}$ for subkey candidates k' $(\neq k)$.

If the subkey k is the right one, then about c ciphertext pairs are expected to pass the β test (see Equation (1) for the β test) due to our assumed related-key differential characteristic of E^0. Otherwise, the expected number of ciphertext pairs passing the β test is about $c \cdot p^{-1} \cdot 2^{-n}$, for the β test has an n-bit filtering condition. Since $p > 2^{-n}$, the expected number of suggested ciphertext pairs for each wrong subkey is less than that for the right subkey. The success rate of this attack depends on the constant c and the number of subkey candidates for E^f. If we exploit an appropriate threshold to keep a portion of the subkey candidates for E^f instead of outputting the subkey with the maximal number T_k of hits (for instance, in Step 3, we can keep all the subkeys that suggest more than $c/2$ ciphertext pairs), then we can sieve many of wrong subkeys for E^f with a higher success rate.[2]

3 DESCRIPTION OF CIKS-128 AND CIKS-128H

In this section, we first introduce some notations and controlled permutations which are the components of CIKS-128 and CIKS-128H. Second, we give descriptions of CIKS-128 and CIKS-128H.

[2] In our attacks on CIKS-128 and CIKS-128H, we directly retrieve subkey bits of E^f using ciphertext pairs remained after a filtration step, since all the filtered corresponding plaintext pairs are expected to be right pairs following our related-key differential characteristics. This is described it in Section 6.

The following notations are used throughout the paper. A bit index will be numbered from left to right, starting with bit 1. If $P = (p_1, p_2, \ldots, p_n)$ then p_1 is the most significant bit and p_n is the least significant bit.

- $e_{i,j}$: a binary string in which the i^{th} and j^{th} bits are one and the others are zeroes, e.g., $e_{1,3} = (1, 0, 1, \ldots, 0)$.
- $\oplus$: bitwise-XOR operation
- $\lll$ ($\ggg$): left (right) cyclic rotation
- $\cap$: logical AND

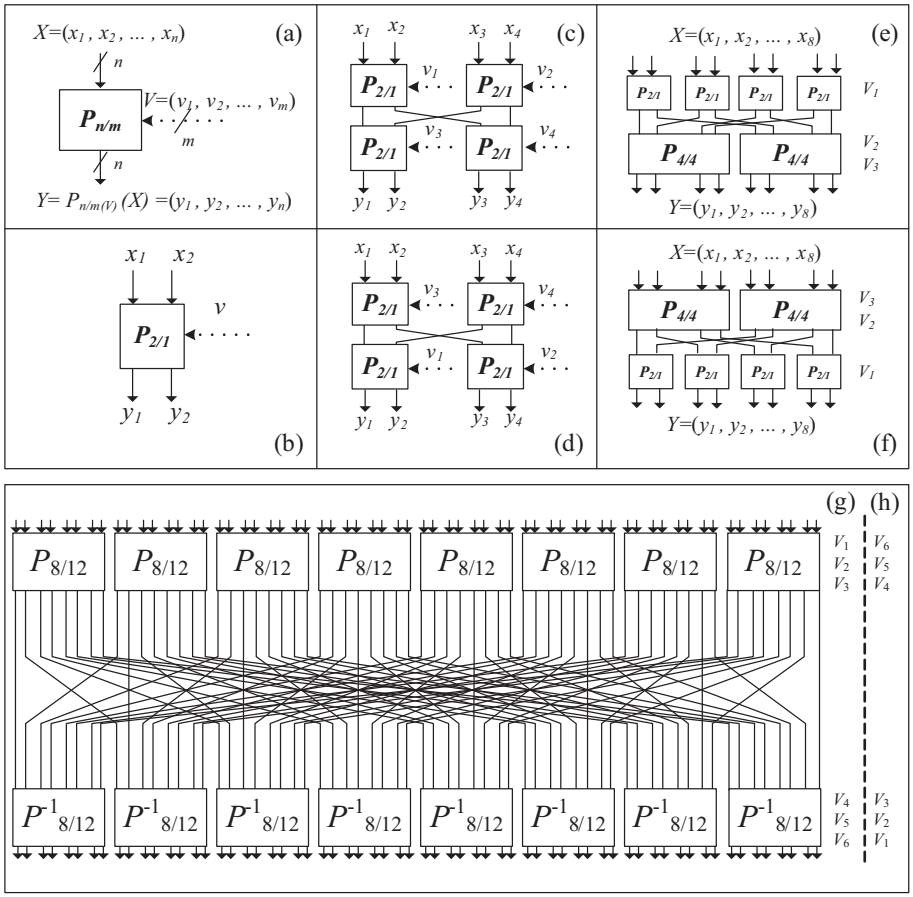


Fig. 1. a) $P_{n/m}$, b) $P_{2/1}$, c) $P_{4/4}$, d) $P_{4/4}^{-1}$, e) $P_{8/12}$, f) $P_{8/12}^{-1}$, h) $P_{64/192}$, g) $P_{64/192}^{-1}$

The DDP can be performed with controlled permutation (CP) boxes, which are defined as follows.

Definition 1. Let $F(X,V)$ be a function $F : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}^n$. F is called a CP-box, if $F(X,V)$ is a bijection for any fixed V.

We denote the above CP-box $F(X,V)$ by $P_{n/m}$, performing permutations on n-bit binary vectors X depending on some controlling m-bit vector V. It is constructed by using switching elements $P_{2/1}$ as elementary building blocks performing controlled transposition of two input bits x_1 and x_2. The $P_{2/1}$-box is controlled by one bit v and outputs two bits y_1 and y_2, where $y_1 = x_{1+v}$ and $y_2 = x_{2-v}$, i.e., if $v = 1$, it swaps two input bits, otherwise (if $v = 0$), it does not.

The $P_{n/m}$-box can be represented as a superposition of the operations performed on bit sets:

$$P_{n/m} = L^{V_1} \circ \pi_1 \circ L^{V_2} \circ \pi_2 \circ \ldots \circ \pi_{s-1} \circ L^{V_s} \tag{2}$$

where L is an active layer composed of $\frac{n}{2}$ $P_{2/1}$ parallel elementary boxes, $V_1, V_2, \ldots, V_s$ are controlling vectors of the active layers from 1 to $s = \frac{2m}{n}$, and $\pi_1, \pi_2, \ldots, \pi_{s-1}$ are fixed permutations (see Figure 1). Figure 1-(g) and -(h) show the structures of $P_{64/192}$ and $P^{-1}_{64/192}$)used in CIKS-128. Due to the symmetric structure and the mutual inverses, $P_{n/m}$ and $P^{-1}_{n/m}$ differ only with the distribution of controlling bits over the boxes $P_{2/1}$, e.g., $P^{V}_{64/192}$ and $P^{V'}_{64/192}$ are mutually inverse when $V = (V_1, V_2, \ldots, V_6)$ and $V' = (V_6, V_5, \ldots, V_1)$, where $|V_i| = 32$ bits.

However, there exists a security problem in the use of the CP-boxes as a cryptographic primitive, which is the linearity of the sum of all the input and output bits of the CP-boxes. This property is defined by the linearity of the elementary building block $P_{2/1}$: $y_1 + y_2 = x_1 + x_2$. In order to thwart this linearity of $P_{2/1}$-box, new non-linear controlled element (NCE), $R_{2/1}$ has been proposed. It is also controlled by one bit v and outputs two bits y_1 and y_2, where $y_1 = f_1(x_1, x_2, v) = x_1 v \oplus x_2 v \oplus x_2$ and $y_2 = f_1(x_1, x_2, v) = x_2 v \oplus x_1$. Similarly, $R_{n/m}$ are constructed as a superposition of $R_{2/1}$ like Equation (2). The only difference between $P_{n/m}$ and $R_{n/m}$ is the used standard elementary box. The $R_{n/m}$ and $R^{-1}_{n/m}$ are used in the round function of CIKS-128H.

We are now ready to describe the block ciphers CIKS-128 [3] and CIKS-128H [20]. These ciphers use the same iterative structure and are composed of the initial transformation (IT), e-dependent round function $Crypt^{(e)}$, and the final transformation (FT) where $e = 0$ ($e = 1$) denotes encryption (decryption) mode as follows:

1. An input block is divided into two subblocks L and R.
2. Perform initial transformation: $L_0 = L \oplus O_3$ and $R_0 = R \oplus O_4$, where O_3 and O_4 are subkeys;
3. For $j = 1$ to $r - 1$ do: $\circ$ $(L_j, R_j) := Crypt^{(e)}(L_{j-1}, R_{j-1}, Q^{(e)}_j)$, where $Q^{(e)}_j$ is the j^{th} round key; $\circ$ Swap the data subblocks: $T = R_j$, $R_j = L_j$, $L_j = T$;
4. $j = r$ do: $(L_r, R_r) := Crypt^{(e)}(L_{r-1}, R_{r-1}, Q^{(e)}_r)$;

5. Perform final transformation: $C_L = L_r \oplus O_1$ and $C_R = R_r \oplus O_2$, where O_1 and O_2 are subkeys;
6. Return the ciphertext block $C = (C_L, C_R)$.

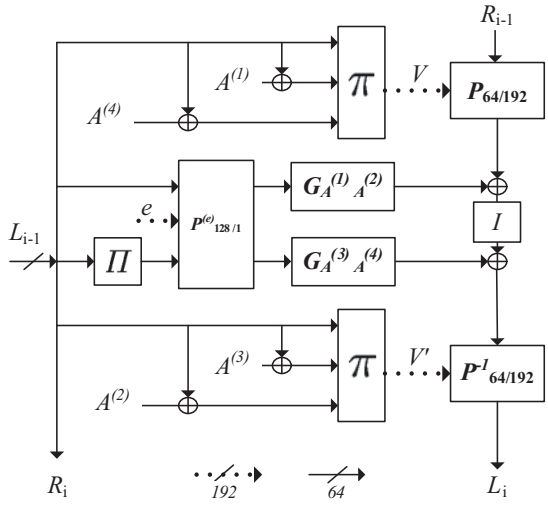


Fig. 2. Crypt$^{(e)}$ of CIKS-128

CIKS-128 [3] is the 12-round iterated block cipher with an 128-bit input and a 256-bit secret key. The $Crypt^{(e)}$ used in CIKS-128 is composed of two fixed permutations π, Π, a permutational involution I, a nonlinear operation G, and two CP-boxes $P^{(V)}_{64/192}$, $(P^{-1}_{64/192})^{(V')}$ (see Figure 2).

Permutation Π contains four cycles of the length 16 represented as follows:

$$(1, 50, 9, 42, 17, 34, 25, 26, 33, 18, 41, 10, 49, 2, 57, 58)$$
$$(3, 64, 43, 24, 19, 48, 59, 8, 35, 32, 11, 56, 51, 16, 27, 40)$$
$$(4, 7, 28, 47, 52, 23, 12, 63, 36, 39, 60, 15, 20, 55, 44, 31)$$
$$(5, 14, 13, 6, 21, 62, 29, 54, 37, 46, 45, 38, 53, 30, 61, 22).$$

Another permutation π forms the control vectors V and V' using three 64-bit input values for the $P_{64/192}$ and $P^{-1}_{64/192}$-box respectively. Let us consider formation of the vector $V = (V_1, V_2, V_3, V_4, V_5, V_6) = \pi(L, L', L'')$ where $V_i (1 \leqq i \leqq 6) \in \{0,1\}^{32}$, $L = (l_1, \ldots, l_{64}) \in \{0,1\}^{64}$: the left input of round function, $(L', L'') = ((l'_1, \ldots, l'_{64}), (l''_1, \ldots, l''_{64})) \in \{(L \oplus A^{(1)}, L \oplus A^{(4)}),\ (L \oplus A^{(3)}, L \oplus A^{(2)})\}$, $A^{(i)} (1 \leqq i \leqq 4) \in \{0,1\}^{64}$: the round keys.

The correspondence between bits of the controlling vector V and elementary switching boxes $P_{2/1}$ of $P^V_{64/192}$-box is given in Table 2, where rows indicate active

layers and numbers correspond to indices of the bits of L, L', L''. The rows corresponding to vectors V_1 and V_4 indicate bits of L. The rows corresponding to vectors V_2 and V_5 indicate bits of L'. The rows corresponding to vectors V_3 and V_6 indicate bits of L''.

For example, according to the table, we have: $V_3 = (l''_{13}, \ldots, l''_{32}, l''_1, \ldots, l''_{12}, l''_{10}, l''_{11}, l''_9)$, $V_4 = (l_{33}, l_{34}, \ldots, l_{64})$. The vector V' is formatted in similar way.

The permutational involution I in one round between $P_{64/192}$ and $P^{-1}_{64/192}$ is performed as follows: $Y = (Y_1, Y_2, \ldots, Y_8) = I(X_1, X_2, \ldots, X_8)$, where $Y_1 = X_6^{\lll 4}$, $Y_2 = X_5^{\lll 4}$, $Y_3 = X_4^{\lll 4}$, $Y_4 = X_3^{\lll 4}$, $Y_5 = X_2^{\lll 4}$, $Y_6 = X_1^{\lll 4}$, $Y_7 = X_8^{\lll 4}$, $Y_8 = X_7^{\lll 4}$.

The transformation $G(L, A', A'') = W$ defining the operation $G_{(A',A'')}(L)$ is described as follows:

$$\begin{aligned} W \;=\; & L_0 \oplus A'_0 \oplus (L_1 \cap A''_0) \oplus (L_2 \cap L_5) \oplus (L_6 \cap A'_1) \\ & \oplus (A''_1 \cap A'_2) \oplus (L_4 \cap L_3) \oplus (L_1 \cap L_6 \cap L_4) \\ & \oplus (L_2 \cap L_6 \cap A''_1) \oplus (L_1 \cap A''_1 \cap L_2 \cap L_4), \end{aligned}$$

where $\forall i \in \{0, 1, 2\}$, $\forall j \in \{0, 1, \ldots, 6\}$, the binary vectors L_j and A_i are defined as: $L_0 = L$, $L_1 = (1, l_1, \ldots, l_{63})$, …, $L_j = (1, \ldots, 1, l_1, \ldots, l_{64-j})$, $A_0 = A$, $A_1 = (1, a_1, \ldots, a_{63})$, $A_2 = (1, 1, a_1, \ldots, a_{62})$ ($A = A'$ or A'').

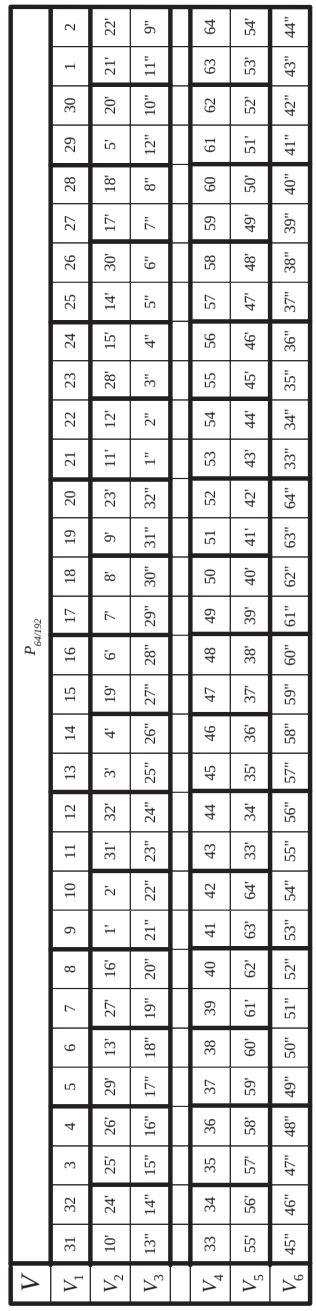
The additional box $P^{(e)}_{128/1}$ does not work in the encryption procedure where $(e) \in \{0, 1\}$ is a parameter defining encryption ($e = 0$) or decryption ($e = 1$) mode. However, it swaps two 64-bit input values in the decryption procedure.

The key schedule of CIKS-128 is very simple. To generate 12 e-dependent round keys $Q^{(e)}_j = \left(A^{(1)}_j, A^{(2)}_j, A^{(3)}_j, A^{(4)}_j\right)$ $(1 \le j \le 12)$, the master key sequences (K_1, K_2, K_3, K_4) are rearranged as specified in Table 3 in which $O_i = K_i$ if $e = 0$, $O_1 = K_3$, $O_2 = K_4$, $O_3 = K_1$, $O_4 = K_2$ if $e = 1$ where $|K_i| = 64$.

To reduce the number of rounds and improve the security of CIKS-128, the block cipher CIKS-128H, which is a modified version of CIKS-128, is introduced in [20]. The differences between them are the DDP-boxes, the key schedule, and the number of rounds. First, the CP-boxes $P_{n/m}$ and $P^{-1}_{n/m}$ in CIKS-128 are replaced by the NCE $R_{n/m}$ and $R^{-1}_{n/m}$ in CIKS-128H, which are constructed by replacing all elementary switching elements $P_{2/1}$ with CEs $R_{2/1}$ that are a nonlinear cryptographic primitive with a minimum size. Second, CIKS-128H uses 8 rounds less than CIKS-128, since the differential properties of $R_{2/1}$ are better than those of $P_{2/1}$. Finally, CIKS-128H uses a modified key schedule in which the subkeys are generated as depicted in Table 4.

4 PROPERTIES OF CIKS-128 AND CIKS-128H

In this section, we describe some properties for components of $\mathrm{Crypt}^{(e)}$ of CIKS-128 and CIKS-128H, which allow us to construct strong related key differential characteristics. To begin with, we present several basic properties of the controlled



V	$P_{64/192}$																															
	31	32	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	1	2
V_1	10'	24'	25'	26'	29'	13'	27'	16'	1'	2'	31'	32'	3'	4'	19'	6'	7'	8'	9'	23'	11'	12'	28'	15'	14'	30'	17'	18'	5'	20'	21'	22'
V_2	13''	14''	15''	16''	17''	18''	19''	20''	21''	22''	23''	24''	25''	26''	27''	28''	29''	30''	31''	32''	1''	2''	3''	4''	5''	6''	7''	8''	12''	10''	11''	9''
V_3																																
	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64
V_4	55'	56'	57'	58'	59'	60'	61'	62'	63'	64'	33'	34'	35'	36'	37'	38'	39'	40'	41'	42'	43'	44'	45'	46'	47'	48'	49'	50'	51'	52'	53'	54'
V_5	45''	46''	47''	48''	49''	50''	51''	52''	53''	54''	55''	56''	57''	58''	59''	60''	61''	62''	63''	64''	33''	34''	35''	36''	37''	38''	39''	40''	41''	42''	43''	44''
V_6																																

Table 2. Distribution of the controlling bits in $P_{64/192}$

j	1	2	3	4	5	6	7	8	9	10	11	12
$A^{(1)}{}_j$	O_1	O_4	O_3	O_2	O_1	O_3	O_3	O_1	O_2	O_3	O_4	O_1
$A^{(2)}{}_j$	O_2	O_3	O_4	O_1	O_2	O_4	O_4	O_2	O_1	O_4	O_3	O_2
$A^{(3)}{}_j$	O_3	O_2	O_1	O_4	O_3	O_1	O_1	O_3	O_4	O_1	O_2	O_3
$A^{(4)}{}_j$	O_4	O_1	O_2	O_3	O_4	O_2	O_2	O_4	O_3	O_2	O_1	O_4

Table 3. Key schedule of CIKS-128

j	1	2	3	4	5	6	7	8
$A^{(1)}{}_j$	K_1	K_4	K_3	K_2	K_2	K_3	K_4	K_1
$A^{(2)}{}_j$	K_2	K_3	K_4	K_1	K_1	K_4	K_3	K_2
$A^{(3)}{}_j$	K_3	K_2	K_1	K_4	K_4	K_1	K_2	K_3
$A^{(4)}{}_j$	K_4	K_1	K_2	K_3	K_3	K_2	K_1	K_4

Table 4. Key schedule of CIKS-128H

elements (Properties 1 and 2), which can induce the properties of components of $\mathrm{Crypt}^{(e)}$.

Property 1. Let $Pr_{(CE)}(\Delta Y/\Delta X, \Delta V)$ be a probability to have the output difference ΔY, where $CE = P_{2/1}$ or $R_{2/1}$, the input difference is ΔX and the difference at the controlling input is ΔV. Then we have the following:

1. $Pr_{(P_{2/1})}((0,0)/(0,0),0) = Pr_{(P_{2/1})}((1,1)/(1,1),0) = 1$
2. $Pr_{(P_{2/1})}(\Delta Y/(0,0),1) = Pr_{(P_{2/1})}(\Delta Y/(1,1),1) = 2^{-1}$ where $\Delta Y = \{(0,0), (1,1)\}$.
3. $Pr_{(P_{2/1})}(\Delta Y/\Delta X, \Delta V) = 2^{-1}$ where $\Delta Y = \{(0,1),(1,0)\}$, $\Delta X = \{(0,1),(1,0)\}$, $\Delta V = \{0,1\}$.
4. $Pr_{(R_{2/1})}((0,0)/(0,0),0) = 1$
5. $Pr_{(R_{2/1})}(\Delta Y/\Delta X, 1) = 2^{-2}$ for any ΔX, ΔY.
6. $Pr_{(R_{2/1})}(\Delta Y_1/(0,1),0) = Pr_{(R_{2/1})}(\Delta Y_2/(1,0),0) = Pr_{(R_{2/1})}(\Delta Y_3/(1,1),0) = 2^{-1}$ where $\Delta Y_1 = \{(0,1),(1,0)\}$, $\Delta Y_2 = \{(0,1),(1,1)\}$, $\Delta Y_3 = \{(1,0),(1,1)\}$.

The above property is also extended into the following properties.

Property 2. Let $Pr_{(CE)}(\Delta Y/\Delta X, \Delta V)$ be a probability to have the output difference ΔY, where $CE = \{P_{n/m}, P^{-1}_{n/m}, R_{n/m}, R^{-1}_{n/m}\}$, the input difference is ΔX and the difference at the controlling input is ΔV. Then we have the following:

1. $Pr_{(P_{n/m})}((e_j)/(e_i), \mathbf{0}) = Pr_{(P^{-1}_{n/m})}((e_j)/(e_i), \mathbf{0}) = 1$ for so me i and j $(1 \le i, j \le n)$

2. $Pr_{(P_{n/m})}((\mathbf{0})/(\mathbf{0}), e_i) = Pr_{(P^{-1}_{n/m})}((\mathbf{0})/(\mathbf{0}), e_i) = 2^{-1}$
3. $Pr_{(R_{n/m})}((\mathbf{0})/(\mathbf{0}), e_i) = Pr_{(R^{-1}_{n/m})}((\mathbf{0})/(\mathbf{0}), e_i) = 2^{-2}$.

The following two properties are very useful in our key recovery attacks on the ciphers.

Property 3. Let $P_{n/m(V)}(X) \oplus P_{n/m(V)}(X \oplus e_i) = e_j$ for some i and j. Then we have the following:

1. If $n = 8$, $m = 12$ then the exact one difference route from e_i to e_j via three $P_{2/1}$-boxes is fixed. It also holds in $P^{-1}_{8/12}$-box.
2. If $n = 64$, $m = 192$ then the exact one difference route from e_i to e_j via six $P_{2/1}$-boxes is fixed. It also holds in $P^{-1}_{64/192}$-box.

Property 4. Let $R_{n/m(V)}(X) \oplus R_{n/m(V)}(X \oplus e_i) = e_j$ for some i and j. Then we have the following:

1. If $n = 8$, $m = 12$, $i = 8$ then the exact one difference route from e_i to e_j via three $R_{2/1}$-boxes is fixed. It also holds in $R^{-1}_{8/12}$-box.
2. If $n = 64$, $m = 192$, $i = 64$ then the exact one difference route from e_i to e_j via six $R_{2/1}$-boxes is fixed. It also holds in $R^{-1}_{64/192}$-box.

For example, consider $i = 8$ and $j = 2$ in Property 3-1. Then, we can exactly know the 3 bits of control vectors $(1, 1, 0)$ corresponding to three elements $P_{2/1}$-boxes of $P_{8/12}$-box with probability 1 (see Figure 3). In Figure 3, the bold line denotes the possible difference route when the input and output differences of $P_{8/12}$ and $P^{-1}_{8/12}$ are fixed.

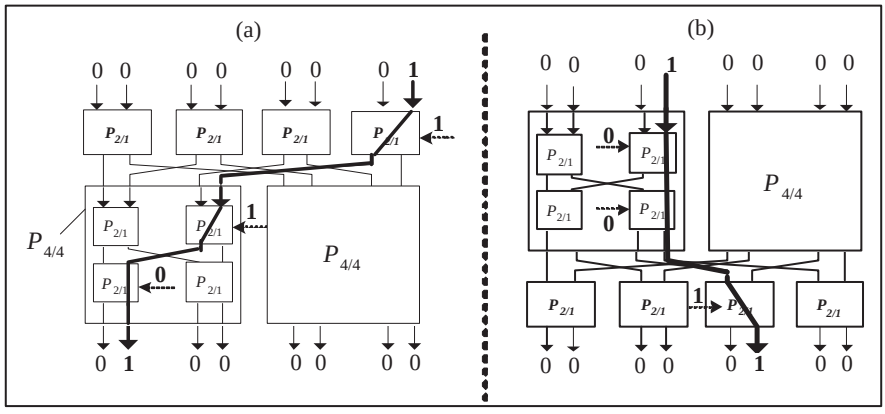


Fig. 3. An example of the difference routes when the input and output differences of $P_{8/12}$ and $P^{-1}_{8/12}$ are fixed

Property 5. The following holds for components of $\mathrm{Crypt}^{(e)}$ of CIKS-128 and CIKS-128H.

1. For the control vector V of $P_{64/192}$-box, $\pi(L, L', L'') \oplus \pi(L, L' \oplus e_{64}, L'') = e_{138}$ and $\pi(L, L', L'') \oplus \pi(L, L', L'' \oplus e_{64}) = e_{180}$. For the control vector V' of $P^{-1}_{64/192}$-box, $\pi(L, L', L'') \oplus \pi(L, L' \oplus e_{64}, L'') = e_{42}$ and $\pi(L, L', L'') \oplus \pi(L, L', L'' \oplus e_{64}) = e_{20}$, where $L, L', L'' \in \{0,1\}^{64}$ and $V, V' \in \{0,1\}^{192}$.
2. If L is a random input and A', A'' are two random round keys, then $G_{A'A''}(L) \oplus G_{A' \oplus e_{64} A'' \oplus e_{64}}(L) = 0$ or e_{64}, i.e., $G_{A' \oplus e_{64} A'' \oplus e_{64}}(L) = 0$ with probability $1/2$ and $G_{A'A''}(L) \oplus G_{A' \oplus e_{64} A'' \oplus e_{64}}(L) = e_{64}$ with probability $1/2$.
3. For any fixed i, j $(1 \le i, j \le 64)$, $\Delta P_{64/192(\Delta V=0)}(\Delta X = e_i) = e_j$ with probability 2^{-6}. (For any fixed i, j there is one difference route in $P_{64/192(V)}(X)$, and this route occurs with probability 2^{-6}.) Similarly, it also holds in $P^{-1}_{64/192}$.
4. For any fixed j $(1 \le i, j \le 64)$, $\Delta R_{64/192(\Delta V=0)}(\Delta X = e_{64}) = e_j$ with probability 2^{-6} because for any fixed i, j, there is one difference route from e_{64} to e_j in $R_{64/192(V)}(X)$, and this route occurs with probability 2^{-6} by Properties 1-6 and 4. Similarly, it also holds in $R^{-1}_{64/192}$.

5 RELATED-KEY DIFFERENTIAL CHARACTERISTICS OF CIKS-128 AND CIKS-128H

We construct related-key differential characteristics for CIKS-128 and CIKS-128H using the properties mentioned in the previous subsection.

5.1 Related-Key Differential Characteristic of CIKS-128

As stated above, the key schedule of CIKS-128 is very simple and there are many useful properties of $P_{64/192}$ and $P^{-1}_{64/192}$ which allow us to construct good related-key differential characteristics with high probability.

In this subsection, we show how to construct full-round related-key differential characteristics for CIKS-128. We consider the situation that we encrypt plaintexts P and P' under an unknown key K and an unknown related-key K' such that $P \oplus P' = (e_{64}, e_{64})$ and $K \oplus K' = (0, 0, e_{64}, e_{64})$, respectively. Then we can obtain 64 desired full-round related-key differential characteristics $(e_{64}, e_{64}) \to (0, e_j)$ $(1 \le j \le 64)$ with the same probability of 2^{-42}, as depicted in Table 5.

Since we consider a related-key pair (K, K') satisfying $K \oplus K' = (0, 0, e_{64}, e_{64})$, we know the difference form of each round key is satisfied with $RK = (0, 0, e_{64}, e_{64})$ or $RK = (e_{64}, e_{64}, 0, 0)$ (Table 3). Now, according to the condition of RK, we describe one round differential characteristic of $\mathrm{Crypt}^{(e)}$ used in our attack.

C1: $RK = (0, 0, e_{64}, e_{64})$

If the input difference of $\mathrm{Crypt}^{(e)}$ in CIKS-128 is zero, then by Property 5-1, the output difference of the first π is e_{180} with probability 1. Thus, by Property 2, the

output difference of $P_{64/192}$ is zero with probability S ($= 2^{-1}$) because the input and controlled vector differences are 0 and e_{180}, respectively. Since the input and round key differences of the second G are 0 and (e_{64}, e_{64}), respectively, the corresponding output difference of the second G is 0 with probability T ($= 2^{-1}$) by Property 5-2. Similarly, the output difference of the second π is e_{42} and the output difference of $P^{-1}_{64/192}$ is 0 with probability U ($= 2^{-1}$). Hence if the input difference of $\mathrm{Crypt}^{(e)}$ is 0 under $RK = (0, 0, e_{64}, e_{64})$ then the corresponding output difference of $\mathrm{Crypt}^{(e)}$ is 0 with probability 2^{-3}.

C2: $RK = (e_{64}, e_{64}, 0, 0)$

Similarly, we can check that the case *C*2 also holds with probability 2^{-3}.

We alternatively use *C1* and *C2* to construct the first 11 rounds of our differential characteristics (see Table 5). In the last round, however, we use a bit different characteristic from *C1* and *C2* for our key recovery attack. In Table 5, the case *C1'* means that the output difference of the second G in the last round is not zero but e_{64} with probability 2^{-1}, and then by Property 2-2, Property 5-3, and Property 5-2, we have $2^{-7}(= U)$ in the last round. See Figure 4.

R (i)	ΔRI^i	ΔRK^i	$S/T/U$	Pr.	Ca.
IT	(e_{64}, e_{64})	(e_{64}, e_{64})	·	1	·
1	$(0, 0)$	$(0, 0, e_{64}, e_{64})$	$2^{-1}/2^{-1}/2^{-1}$	2^{-3}	*C1*
2	$(0, 0)$	$(e_{64}, e_{64}, 0, 0)$	$2^{-1}/2^{-1}/2^{-1}$	2^{-3}	*C2*
3	$(0, 0)$	$(e_{64}, e_{64}, 0, 0)$	$2^{-1}/2^{-1}/2^{-1}$	2^{-3}	*C2*
4	$(0, 0)$	$(0, 0, e_{64}, e_{64})$	$2^{-1}/2^{-1}/2^{-1}$	2^{-3}	*C1*
5	$(0, 0)$	$(0, 0, e_{64}, e_{64})$	$2^{-1}/2^{-1}/2^{-1}$	2^{-3}	*C1*
6	$(0, 0)$	$(e_{64}, e_{64}, 0, 0)$	$2^{-1}/2^{-1}/2^{-1}$	2^{-3}	*C2*
7	$(0, 0)$	$(e_{64}, e_{64}, 0, 0)$	$2^{-1}/2^{-1}/2^{-1}$	2^{-3}	*C2*
8	$(0, 0)$	$(0, 0, e_{64}, e_{64})$	$2^{-1}/2^{-1}/2^{-1}$	2^{-3}	*C1*
9	$(0, 0)$	$(0, 0, e_{64}, e_{64})$	$2^{-1}/2^{-1}/2^{-1}$	2^{-3}	*C1*
10	$(0, 0)$	$(e_{64}, e_{64}, 0, 0)$	$2^{-1}/2^{-1}/2^{-1}$	2^{-3}	*C2*
11	$(0, 0)$	$(e_{64}, e_{64}, 0, 0)$	$2^{-1}/2^{-1}/2^{-1}$	2^{-3}	*C2*
12	$(0, 0)$	$(0, 0, e_{64}, e_{64})$	$2^{-1}/2^{-1}/2^{-7}$	2^{-9}	*C1'*
FT	$(0, e_j)$	$(0, 0)$	·	1	·
Outp.	$(0, e_j)$	·	·	·	·
Total	·	·	·	2^{-42}	·

- $1 \le j \le 64$: fixed value, Outp.:Output,
- Pr.:Probability, Ca.:Case

Table 5. Related-key differential characteristic of CIKS-128

5.2 Related-Key Differential Characteristic of CIKS-128H

The main difference between CIKS-128 and CIKS-128H is the DDP-box, i.e., CIKS-128H uses $R_{n/m}$-boxes instead of $P_{n/m}$-boxes. So, we can similarly present related

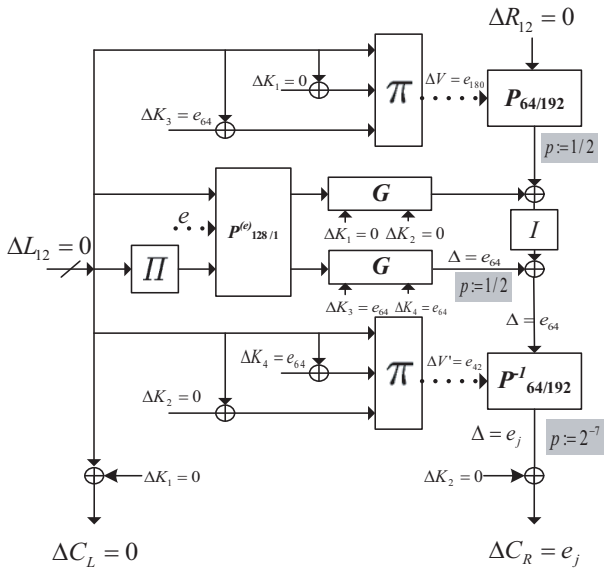


Fig. 4. Propagation of the difference in the last round

key differential characteristics on the full-round CIKS-128H. We first consider the situation that we encrypt plaintexts P and P' under an unknown key K and an unknown related-key K' such that $P \oplus P' = (e_{64}, e_{64})$ and $K \oplus K' = (0, 0, e_{64}, e_{64})$, respectively. Then, we similarly construct the one-round iterative related key differential characteristics under the same condition represented in the previous Subsection 5.1. The probabilities of *C1* and *C2* are replaced with 2^{-5} by Property 1-5, Property 2-3, and Property 5-2 and then we alternatively use *C1* and *C2* to construct the first 7 rounds of our differential characteristics (see Table 6). In the last round, however, we use the case *C1'* which means that the output difference of the second G in the last round is not zero but e_{64} with probability 2^{-1} Property 5-2, and then by Property 2-3, and Property 5-4, we have the probability of 2^{-8} $(= U)$ in the last round.

Therefore, we can obtain 64 desired full-round related-key differential characteristics $(e_{64}, e_{64}) \rightarrow (0, e_j)$ $(1 \leq j \leq 64)$ with probability 2^{-46}, as depicted in Table 6.

6 KEY RECOVERY ATTACKS ON CIKS-128 AND CIKS-128H

We now present key recovery attacks on CIKS-128 and CIKS-128H using our related-key differential characteristics.

To begin with, we encrypt 2^{43} plaintext pairs $P = (P_L, P_R)$ and $P' = (P_L \oplus e_{64}, P_R \oplus e_{64})$ under an unknown key $K = (K_1, K_2, K_3, K_4)$ and an unknown related-key $K' = (K_1, K_2, K_3 \oplus e_{64}, K_4 \oplus e_{64})$, respectively, and then get the 2^{43} corre-

R (i)	ΔRI^i	ΔRK^i	$S/T/U$	Pr.	Ca.
IT	(e_{64}, e_{64})	(e_{64}, e_{64})	·	1	·
1	$(0, 0)$	$(0, 0, e_{64}, e_{64})$	$2^{-2}/2^{-1}/2^{-2}$	2^{-5}	*C1*
2	$(0, 0)$	$(e_{64}, e_{64}, 0, 0)$	$2^{-2}/2^{-1}/2^{-2}$	2^{-5}	*C2*
3	$(0, 0)$	$(e_{64}, e_{64}, 0, 0)$	$2^{-2}/2^{-1}/2^{-2}$	2^{-5}	*C2*
4	$(0, 0)$	$(0, 0, e_{64}, e_{64})$	$2^{-2}/2^{-1}/2^{-2}$	2^{-5}	*C1*
5	$(0, 0)$	$(0, 0, e_{64}, e_{64})$	$2^{-2}/2^{-1}/2^{-2}$	2^{-5}	*C1*
6	$(0, 0)$	$(e_{64}, e_{64}, 0, 0)$	$2^{-2}/2^{-1}/2^{-2}$	2^{-5}	*C2*
7	$(0, 0)$	$(e_{64}, e_{64}, 0, 0)$	$2^{-2}/2^{-1}/2^{-2}$	2^{-5}	*C2*
8	$(0, 0)$	$(0, 0, e_{64}, e_{64})$	$2^{-2}/2^{-1}/2^{-8}$	2^{-11}	*C1'*
FT	$(0, e_j)$	$(0, 0)$	·	1	·
Outp.	$(0, e_j)$	·	·	·	·
Total	·	·	·	2^{-46}	·

- $1 \leq j \leq 64$: fixed value, Outp.:Output,
- Pr.:Probability, Ca.:Case

Table 6. Related-key differential characteristic of CIKS-128H

sponding ciphertext pairs $C = (C_L, C_R)$ and $C' = (C'_L, C'_R)$, i.e., $E_K(P) = C$ and $E_{K'}(P') = C'$, where E is the block cipher CIKS-128. Since our full-round related-key differential characteristic of CIKS-128 has a probability of 2^{-42}, we expect about 1 ciphertext pair (C, C') such that $C \oplus C' = (0, e_j)$ for each j $(1 \leq j \leq 64)$. According to our differential trails described in Table 5, we can deduce that the j^{th} one-bit difference in such (C, C') is derived from the 64^{th} input difference of $P^{-1}_{64/128}$ in the last round (Figure 5). Recall that there is a unique difference route when the input and output differences with hamming weight 1 are determined in $P_{64/192}$ and $P^{-1}_{64/192}$. Then, we can extract 6 bits of control vectors and the corresponding key bits by using Property 3. Based on this idea we can devise a key recovery attack on the full-round CIKS-128.

1. For CIKS-128, prepare 2^{43} plaintext pairs (P_i, P'_i), $i = 1, \ldots, 2^{43}$, which have the (e_{64}, e_{64}) difference. All P_i are encrypted using a master key K and all P'_i are encrypted using a master key K' where K and K' have the $(0, 0, e_{64}, e_{64})$ difference. Encrypt each plaintext pair (P_i, P'_i) to get the corresponding ciphertext pair (C_i, C'_i).
2. Check that $C_i \oplus C'_i = (0, e_j)$ for each i and j $(1 \leq j \leq 64)$.
3. For each ciphertext pair (C_i, C'_i) passing Step 2, extract some bits of control vector by chasing a difference route between this PBO and the position of the 64^{th} input bit in $P^{-1}_{64/192}$ (see Figure 5). Then find the corresponding bits of K_1, $K_1 \oplus K_2$, and $K_1 \oplus K_3$. Note that the controlled vector V' of $P^{-1}_{64/192}$ in the last round is formatted with $C_L \oplus K_1$, $C_L \oplus K_1 \oplus K_2$, and $C_L \oplus K_1 \oplus K_3$.

The data complexity of this attack is 2^{44} related-key chosen plaintexts. The time complexity of Step 1 is 2^{44} full-round CIKS-128 encryptions and the time complexity

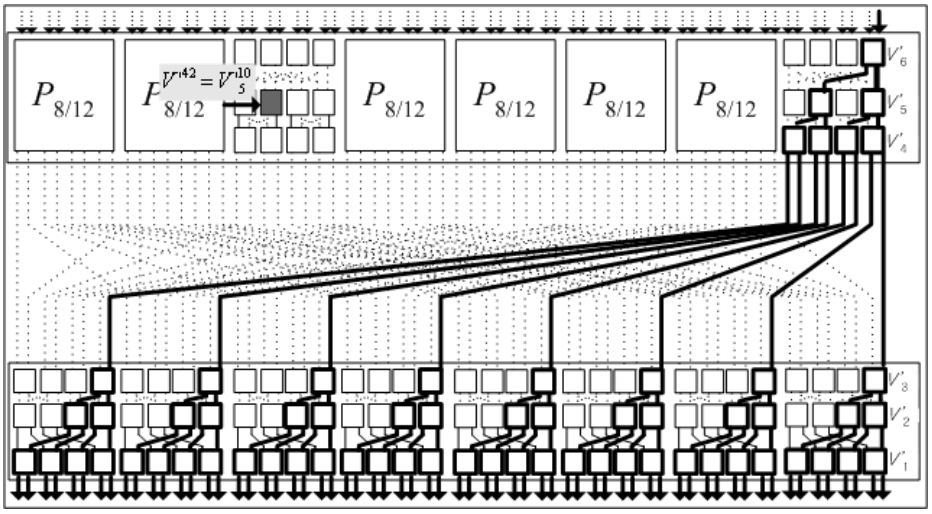


Fig. 5. The possible routes of the 64$^{\text{th}}$ difference of $P^{-1}_{64/192}$

of Steps 2 and 3 is much less than that of Step 1. By our related-key differential characteristics each ciphertext pair can pass Step 2 with probability at least 2^{-42} and thus the expectation of ciphertext pair with the $(0, e_j)$ difference that pass this test is at least 2. Therefore we have at least one ciphertext pair with the $(0, e_j)$ difference for each j $(1 \leq j \leq 64)$. Thus we can retrieve 56 bits of information of keys in the lower layer of $P^{-1}_{64/192}$ and 7 bits of information of keys in the upper layer of $P^{-1}_{64/192}$ with data and a time complexities of 2^{44}.

Similarly, if we replace 2^{43} related-key plaintext pairs in the above algorithm by 2^{47} related-key plaintext pairs then we can also succeed in finding 63 bits of information of keys for CIKS-128H with a time complexity of 2^{48} encryptions. Furthermore, these attacks can be simply extended to retrieve the whole of master key pair (K, K') by performing an exhaustive search for the remaining keys.

7 CONCLUSION

CIKS-128, and CIKS-128H have been designed suitable for intelligent multimedia and ubiquitous computing systems which require lower power and high-speed performance. They are also considerably resistant against conventional attacks such as the differential attack and the linear attack. However, they use a weak diffusion, a weak non-linear operation, and a very simple key schedule, leading to full-round related-key differential attacks. According to our results, the full-round CIKS-128 and CIKS-128H can be broken by 2^{44} and 2^{48} data/time complexities, respectively. Our results indicate that CIKS-128 and CIKS-128H are almost in practical attack bounds.

Acknowledgement

This work was supported by Kyungnam University Foundation Grant, 2010

REFERENCES

[1] Biham, E.: New Types of Cryptanalytic Attacks Using Related Keys. Advances in Cryptology – Eurocrypt '03, LNCS 765, pp. 398–409, Springer-Verlag, 1994; Journal of Crytology, Vol. 7, No. 4, pp. 229–246, 1994.

[2] Biham, E.—Shamir, A.: Differential Cryptanalysis of the Data Encryption Standard. ISBN: 0-387-97930-1, 3-540-97930-1, 1993.

[3] Goots, N. D.—Izotov, B. V.—Moldovyan, A. A.—Moldovyan, N. A.: Modern cryptography: Protect Your Data with Fast Block Ciphers. Wayne, A-LIST Publish., 2003.

[4] Goots, N. D.—Izotov, B. V.—Moldovyan, A. A.—Moldovyan, N. A.: Fast Ciphers for Cheap Hardware: Differential Analysis of SPECTR-H64. MMM-ACNS '03, LNCS 2776, pp. 449–452, Springer-Verlag 2003.

[5] Goots, N. D.—Moldovyan, N. A.—Moldovyanu, P. A.—Summerville, D. H.: Fast DDP-Based Ciphers: From Hardware to Software. 46th IEEE Midwest International Symposium on Circuits and Systems 2003.

[6] Izotov, B. V.—Moldovyan, A. A.—Moldovyan, N. A.: Fast Encryption Algorithm Spectr-H64. MMM-ACNS '01, LNCS 2052, pp. 275–286, Springer-Verlag 2001.

[7] Kavut, S.—Yücel, M. D.: Slide Attack on Spectr-H64. INDOCRYPT '02, LNCS 2551, pp. 34–47, Springer-Verlag 2002.

[8] Kelsey, J.—Schneier, B.—Wagner, D.: Key Schedule Cryptanalysis of IDEA, G-DES, GOST, SAFER, and Triple-DES. Advances in Cryptology – CRYPTO '96, LNCS 1109, pp. 237–251, Springer-Verlag 1996.

[9] Kelsey, J.—Schneier, B.—Wagner, D.: Related-Key Cryptanalysis of 3-WAY, Biham-DES, CAST, DES-X, NewDES, RC2, and TEA. ICICS '97, LNCS 1334, pp. 233–246, Springer-Verlag 1997.

[10] Ko, Y.—Lee, C.—Hong, S.—Lee, S.: Related Key Differential Cryptanalysis of Full-Round SPECTR-H64 and CIKS-1. ACISP 2004, LNCS 3108, pp. 137–148, Springer-Verlag 2004.

[11] Ko, Y.—Lee, C.—Hong, S.—Sung, J.—Lee, S.: Related-Key Attacks on DDP based Ciphers: CIKS-128 and CIKS-128H. Indocrypt 2004, LNCS 3348, pp. 191–205, Springer-Verlag 2004.

[12] Knudsen, L. R.: Cryptanalysis of LOKI91. Advances in Cryptology – Proceedings of AUSCRYPT 1992, LNCS 718, pp. 196–208, Springer-Verlag 1993.

[13] Lee, C.—Hong, D.—Lee, S.—Lee, S.—Yang, H.—Lim, J.: A Chosen Plaintext Linear Attack on Block Cipher CIKS-1. ICICS 2002, LNCS 2513, pp. 456–468, Springer-Verlag 2002.

[14] Lee, C.—Kim, J.—Hong, S.—Sung, J.—Lee, S.: Related-Key Differential Attacks on Cobra-S128, Cobra-F64a, and Cobra-F64b. MYCRYPT 2005, LNCS 3715, pp. 245–263, Springer-Verlag 2005.

[15] Lee, C.—Kim, J.—Sung, J.—Hong, S.—Lee, S.: Related-Key Differential Attacks on Cobra-H64 and Cobra-H128. Tenth IMA International Conference On Cryptography and Coding (CCC 2005), LNCS 3796, pp. 201–219, Springer-Verlag 2005.

[16] Matsui, M.: Linear Cryptanalysis Method for DES Cipher. Advances in Cryptology – EUROCRYPTO '93, LNCS 765, pp. 386–397, Springer-Verlag 1993.

[17] Moldovyan, A. A.—Moldovyan, N. A.: A Cipher Based on Data-Dependent Permutations. Journal of Cryptology, Vol. 15, 2002, No. 1, pp. 61–72.

[18] Sklavos, N.—Moldovyan, N. A.–Koufopavlou, O.: High Speed Networking Security: Design and Implementation of Two New DDP-Based Ciphers. Mobile Networks and Applications – MONET, Kluwer Academic Publishers, Vol. 25, 2005, Issue 1-2, pp. 219–231.

[19] Sklavos, N.—Moldovyan, N. A.—Koufopavlou, O.: Encryption and Dada Dependent Permutations: Implementation Cost and Performance Evaluation. MMM-ACNS 2003, LNCS 2776, pp. 337–348, Springer-Verlag 2003.

[20] Sklavos, N.—Moldovyan, N. A.—Koufopavlou, O.: A New DDP-based Cipher CIKS-128H: Architecture, Design & VLSI Implementation Optimization of CBC-Encryption & Hashing over 1 GBPS, Proceedings of 46th IEEE Midwest Symposium on Circuits & Systems, December 27–30, Cairo, Egypt 2003.

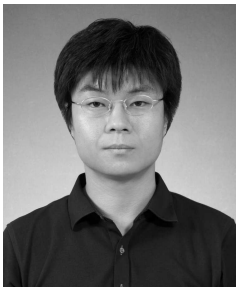
[21] Sklavos, N.—Koufopavlou, O.: Dada Dependent Rotations – A Trustworthy Approach for Future Encryption and Systems/Ciphers: Low Cost and High Performance. Computers and Security, Elsevier Science Journal, Vol. 22, 2003, No. 7.

[22] Phan, R. C. W.—Handschuh, H.: On Related-Key and Collision Attacks: The case for the IBM 4758 Cryptoprocessor. ISC 2004, LNCS 3225, pp. 111–122, Springer-Verlag 2004.

[23] Razali, E.—Phan, R. C. W.: On the Existence of Related-Key Oracles in Cryptosystems Based on Block Ciphers. OTM Workshops 2006, LNCS 4277, pp. 425–438, Springer-Verlag 2006.

Changhoon Lee received his Ph. D. degree in Graduate School of Information Management and Security (GSIMS) from Korea University, Korea. He is now a Professor at the School of Computer Science, Hanshin University, Korea. In scientific activities area, he was the Program Co-Chair of International Conference on Security Technology (SecTech-08), International Symposium on Grid and Distributed Computing (GDC-08), International Conference on Information Security and Assurance (ISA-09), International Symposium on Security and Multimodality in Pervasive Environments (SMPE-09), International Symposium on Ubiquitous Applications & Security Services (UASS-09), and the 2nd International Conference on Computer Science and its Applications (CSA-09). He now serves as a Workshop Chair for HumanCom-10, a finance and registration chair for EMC-10 and FutureTech-10,

and a local chair for ICA3PP-10. He is editorial board member of the International Journal of Information Processing System (JIPS) and International Journal of Information Technology, Communications and Convergence (IJITCC). His current research interests include information security, cryptology, ubiquitous security, digital forensics, context awareness, game, etc. He is a member of the IEEE, IEEE Computer Society, IEEE Communication, IACR, KIISC, KMMS, KICS, and KIIT.



Jongsung Kim received his Bachelor and Master degrees in Mathematics from Korea university, Korea in 2000 and 2002, respectively. He received double Doctoral degrees on "Combined Differential, Linear and Related-Key Attacks on Block Ciphers and MAC Algorithms", completed in November 2006 and February 2007 at the ESAT/COSIC group of Katholieke Universiteit Leuven and at Engineering in Information Security of Korea University, respectively. He had been a Research Professor of Center for Information Security Technologies (CIST) at Korea University, from March 2007 till August 2009. He has been an Assistant Professor of Division of e-Business, Kyungnam University, Korea, since September 2009. His research interests include symmetric cryptosystems, side-channel attacks, ubiquitous computing systems and digital forensics.

Jaechul Sung received his Bachelor, Master and Ph. D. degrees in Mathematics from the Korea University, Seoul, Korea, in 1997, 1999 and 2002, respectively. From 2002 to 2004 he was a senior research engineer in Korea Information Security Agency (KISA). Since 2004, he has been an Associate Professor of the University of Seoul, Seoul, Korea. His current research interest includes cryptography, symmetric cryptosystems, hash functions and MACs.

Yang-Sun Lee received the B. Sc. and M. Sc. degrees in Electrical & Electronic Engineering from Dongshin University and Ph. D. degrees in Department of IT Engineering from Mokwon University in 2001, 2003 and 2007, respectively. He was a Senior Engineer at R & D Center, Fumate Co., Ltd. between 2007 to 2009. Since 2009, he has been working as a Research Professor in Department of Information Communication Engineering at Chosun University. His current research interests include UWB, multimedia communication, network transmission scheme and ubiquitous sensor networks. He is a member of the KICS, KIMICS, KEES, KIIT, KITCS and KONI.

Chang Hoon Lee received a B. Sc. degree in Computer Science from KwangWoon University in 1987, and M. Sc. and Ph. D. degrees in Computer Science from ChungAng University in 1989 and 1998. Since 2002 he has been a Professor of Computer Engineering Department at HanKyong National University, Korea. His research interests include Object-Oriented Design, Formal Specification, and web-services.